NSDE 1: LECTURE 6

TYRONE REES[*]

$$u' = f(t,u), \qquad u(t_0) = u_0.$$

Last time we looked at ways of controlling the step size using adaptive algorithms. This is useful for when we want the error to remain within some tolerance.

Another strategy is to use Runge-Kutta methods of different orders on top of each other. Because of the flexibility allowed in RK methods, it is possible to find sets of two families that evaluate the function at the same points, yet have different orders. For example, consider the method with Butcher table:

$$\begin{array}{c|cccc} 0 & 1 & & & \\ 1/2 & 1/2 & & & \\ 3/4 & 0 & 3/4 & & \\ 1 & 2/9 & 3/9 & 4/9 & \\ \hline & 2/9 & 3/9 & 4/9 & \\ & 11/72 & 30/72 & 40/72 & -9/72 \end{array},$$

which is short-hand for the two methods

$$U_{n+1}^a = U_n + \frac{\Delta t_n}{9}(2k_1 + 3k_2 + 4k_3)$$

(a second order method), and

$$U_{n+1}^b = U_n + \frac{\Delta t_n}{72}(11k_1 + 30k_2 + 40k_3 - 9k_4)$$

(a third order method).

Here we can say that $u(t_{n+1}) = U_{n+1}^a + K_2(\Delta t_n)^2 + O(\Delta t_n)^3$, and also that $u(t_{n+1}) = U_{n+1}^b + O(\Delta t_n)^3$. Since, for small Δt_n, we can assume that the $O(\Delta t_n)^3$ term is negligible, we have that

$$|U_{n+1}^b - U_{n+1}^a| \approx K_2(\Delta t_n)^2.$$

We can then follow the same procedure as before. This scheme (developed by Dogacki and Shampine in 1989) is the basis of the ode23 solver in Matlab.

The first such method to be found was an order five-six pair, discoverd by Fehlberg (working for NASA) in the late sixties. A similar method,

[*] Rutherford Appleton Laboratory, Chilton, Didcot, UK, tyrone.rees@stfc.ac.uk

based on fourth and fifth order methods, was developed by Dormand and Price (1980), and this is the basis for the ode45 solver in matlab. It's Butcher table is:

0	1						
1/5	1/5						
3/10	3/40	9/40					
4/5	44/45	−56/15	32/9				
8/9	19372/6561	−25360/2187	64448/6561	−212/729			
1	9017/3168	−355/33	46732/5247	49/176	−5103/18656		
1	35/384	0	500/1113	125/192	−2187/6784	11/84	
c_r	35/384	0	500/1113	125/192	−2187/6784	11/84	0
d_r	5179/57600	0	7571/16695	393/640	−92097/339200	187/2100	1/40

Symplectic methods

Sometimes, however, it's not (only) the error that we're interested in.

A very important field of study is dynamical systems, in particular *Hamiltonian systems*. A Hamiltonian system consists of $l = 2m$ differential equations:

$$x_i' = \frac{\partial H}{\partial u_i}$$
$$u_i' = -\frac{\partial H}{\partial x_i}$$

for $i = 1, \ldots, l$. The scalar function $H(\mathbf{x}, \mathbf{u})$ is called the *Hamiltonian.* (see B7.1). A typical example is a system of particles; in this case $\mathbf{x}(t)$ are the positions of particles at time t, and $\mathbf{u}(t)$ are their velocities. In this case H is the total energy.

Example

Consider a simple harmonic oscillator. Let x be the postition at time t, and u be it's velocity. The (scaled) system can be written as

$$x'' = -x \Rightarrow x' = u, \; u' = -x.$$

This can be set in Hamiltonian framework by writing the Hamiltonian

$$H = \frac{1}{2}(x^2 + u^2).$$

Note that the Hamiltonian corresponds to the total energy in the system, and so conservation of energy tells us that solution of the dynamical system correspond to the Hamiltonian being constant – in this case, circles in the (x, u) plane.

Let the solution be given by $\mathbf{u} = (x, u)$. Then the Hamiltonian can be written as $H = \frac{1}{2}\mathbf{u}^T\mathbf{u}$. A reasonable question to ask is how this evolves using the numerical scheme.

Explicit Euler for this system looks like

$$\begin{bmatrix}X_{n+1}\\U_{n+1}\end{bmatrix} = \begin{bmatrix}X_n\\U_n\end{bmatrix} + h\begin{bmatrix}U_n\\-X_n\end{bmatrix} = \begin{bmatrix}1 & h\\-h & 1\end{bmatrix}\begin{bmatrix}U_n\\X_n\end{bmatrix}$$

and so

$$\begin{bmatrix}X_{n+1} & U_{n+1}\end{bmatrix}\begin{bmatrix}X_{n+1}\\U_{n+1}\end{bmatrix} = \begin{bmatrix}U_n & X_n\end{bmatrix}\begin{bmatrix}1 & -h\\h & 1\end{bmatrix}\begin{bmatrix}1 & h\\-h & 1\end{bmatrix}\begin{bmatrix}U_n\\X_n\end{bmatrix} = (1+h^2)\begin{bmatrix}X_n & U_n\end{bmatrix}\begin{bmatrix}X_n\\U_n\end{bmatrix}$$

and so

$$H_{n+1} = (1+h^2)H_n.$$

So the Hamiltonian (and hence the energy) grows in time for all time steps.

Now, consider the hybrid Euler scheme

$$\begin{bmatrix}U_{n+1}\\X_{n+1}\end{bmatrix} = \begin{bmatrix}U_n\\X_n\end{bmatrix} + h\begin{bmatrix}-X_n\\U_{n+1}\end{bmatrix}$$

or, in matrix terms:

$$\begin{bmatrix}1 & -h\\0 & 1\end{bmatrix}\begin{bmatrix}X_{n+1}\\U_{n+1}\end{bmatrix} = \begin{bmatrix}1 & 0\\-h & 1\end{bmatrix}\begin{bmatrix}X_n\\U_n\end{bmatrix}$$

This also doesn't leave the Hamiltonian H unaltered, but it does preserve a modified Hamiltonian:

$$\hat{H}(x,u) = \frac{1}{2}(x^2+u^2) - \frac{1}{2}hxu = H(x,u) - \frac{1}{2}hxu.$$

This is called a **symplectic scheme**; while the original Hamiltonian is not preserved, it is recovered in the limit as $h \to 0$.

A common symplectic scheme is the **Stömer-Verlet** scheme. If the system to be solved is

$$\begin{aligned}x' &= u\\u' &= f(x),\end{aligned}$$

then the Stömer-Verlet scheme takes the form:

$$\begin{aligned}U_{n+1/2} &= U_n + \frac{1}{2}hf(X_n)\\X_{n+1} &= X_n + hU_{n+1/2}\\U_{n+1} &= U_{n+1/2} + \frac{1}{2}hf(X_{n+1}),\end{aligned}$$

which, if we eliminate the intermediate value, gives the scheme

$$X_{n+1} = X_n + hU_n + \frac{1}{2}h^2 f(X_n)$$
$$U_{n+1} = U_n + \frac{1}{2}[f(X_n) + f(X_{n+1})]$$

Consider an area in the (x, u) plane, corresponding to a set of initial conditions. Another property of Hamiltonian mechanics is that, as the state of the system progresses, the area remains constant.

Recall that the area of a parallelogram with corners $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ is given by

$$A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

Consider again the dynamical system

$$x' = u \qquad u' = -x.$$

Now, suppose again that we had three initial conditions $(X_0^1, U_0^1), (X_0^2, U_0^2), (X_0^3, U_0^3)$, and evolve these using Euler's method

$$X_n^i = X_n^i + hU_n^i, \qquad U_n^i = U_n^i - hX_n^i,$$

with area

$$A_n = \begin{vmatrix} X_n^1 & U_n^1 & 1 \\ X_n^2 & U_n^2 & 1 \\ X_n^3 & U_n^3 & 1 \end{vmatrix}$$

To make the algebra easier, let's choose initial (X_0, Y_0), $(X_0 + A, Y_0)$ and $(X_0, Y_0 + B)$.

Then after one step of Euler these move to

$$\begin{aligned} &(X_0 + hU_0, U_0 - hX_0), \\ &(X_0 + A + hU_0, U_0 - h(X_0 + A)) \text{ and} \\ &(X_0 + h(U_0 + B), U_0 + B - hX_0) \end{aligned}$$

The area is therefore given by

$$\begin{aligned} A &= \begin{vmatrix} X_0 + hU_0 & U_0 - hX_0 & 1 \\ X_0 + A + hU_0 & U_0 - h(X_0 + A) & 1 \\ X_0 + h(U_0 + B) & U_0 + B - hX_0 & 1 \end{vmatrix} \\ &= \begin{vmatrix} X_0 + hU_0 & U_0 - hX_0 & 1 \\ A & -hA & 0 \\ hB & B & 0 \end{vmatrix} \\ &= (1 + h^2)AB. \end{aligned}$$

Therefore the area increases exponentially as Euler progresses. Symplectic methods preserve the area (see problem sheet 2).